

IN THE CLAIMS:

This listing of the claims replaces all prior versions and listings of the claims in this application.

The text of all pending claims (including any withdrawn claims) is set forth below. Canceled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~strike through~~. The status of each claim is indicated with one of (Original), (Currently amended), (Canceled), (Withdrawn), (Previously presented), (New), and (Not entered).

Please AMEND claims 1, 3, 5-7, 10, and 11 in accordance with the following:

1. (Currently amended) A non-transitory computer-readable storage medium usable with an apparatus comprising a buffer, the non-transitory computer-readable storage medium having recorded thereon:

audio video (AV) data;

a markup document to be preloaded into the buffer of the apparatus to enable the apparatus to reproduce the AV data in an interactive mode selected by a user of the apparatus, wherein the markup document does not comprise the AV data or any other AV data; and

control information providing functionality to enable the apparatus to identify buffering state information of the markup document to be preloaded into the buffer of the apparatus, the buffering state information being used by the apparatus in reproducing the AV data in the interactive mode selected by the user;

wherein:

the control information comprises an application program interface (API) that generates a report signal used to identify a buffering state of the markup document; and

the report signal is used by the apparatus to verify whether the markup document has been successfully preloaded into the buffer, whether the markup document cannot be read due to an error, and whether the markup document is being read.

2. (Canceled)

3. (Currently amended) The non-transitory computer-readable storage medium of claim 1, wherein the API comprises an [obj].isCached(URL, resType) API that generates the report signal, where the URL is a parameter indicating a file path of the markup document, and the resType is a parameter indicating an attribute of the markup document.

4. (Canceled)

5. (Currently amended) The non-transitory computer-readable storage medium of claim 1, wherein the control information further comprises an API that generates a fetch signal used to issue a command to preload the markup document.

6. (Currently amended) The non-transitory computer-readable storage medium of claim 5, wherein the API that generates the fetch signal returns a response indicating whether the command to preload the markup document has been successfully transmitted using the fetch signal.

7. (Currently amended) The non-transitory computer-readable storage medium of claim 1, wherein the control information further comprises an API that is used to determine whether preloading of the markup document is completed.

8.-9. (Canceled)

10. (Currently amended) The non-transitory computer-readable storage medium of claim 1, wherein the interactive mode is a mode in which the AV data is displayed in a display window defined by the markup document;

the apparatus is selectively operable in the interactive mode in which the AV data is displayed in the display window defined by the markup document, and a non-interactive video mode in which the AV data is displayed in the same manner as AV data recorded on a standard DVD; and

the user of the apparatus selects between the interactive mode and the non-interactive video mode.

11. (Currently amended) The non-transitory computer-readable storage medium of claim 1, further having recorded thereon a startup markup document separate from the markup document to be preloaded into the buffer of the apparatus and comprising preloading instructions enabling the apparatus to preload the markup document into the buffer of the apparatus;

wherein the selection of the interactive mode by the user causes the apparatus to read the startup markup document from the non-transitory computer-readable storage medium and execute the preloading instructions to preload the markup document into the buffer of the apparatus.

12.–13. (Canceled)

14. (Previously presented) An apparatus for reproducing audio video (AV) data using a markup document in an interactive mode selected by a user of the apparatus, comprising:
a buffer to buffer the markup document to enable the apparatus to reproduce the AV data in the interactive mode selected by the user; and
a buffer manager to manage the buffer to preload the markup document and output buffering state information of the buffer in response to a report signal, the buffering state information being used by the apparatus in reproducing the AV data in the interactive mode selected by the user;

wherein:

the apparatus generates the report signal using an application program interface (API); and

the report signal is used by the buffer manager to verify whether the markup document has been successfully preloaded into the buffer, whether the markup document cannot be read due to an error, and whether the markup document is being read.

15. (Previously presented) The apparatus of claim 14, further comprising a content decoder to interpret the markup document, and generate the report signal using the API;
wherein the buffer manager informs the content decoder of the buffering state information of the buffer in response to the report signal.

16. (Canceled)

17. (Previously presented) The apparatus of claim 15, wherein the comprises a file path of the markup document and an attribute of the markup document as parameters.

18. (Previously presented) The apparatus of claim 15, wherein the API comprises an [obj].isCached(URL, resType) API that generates the report signal, where the URL is a parameter indicating a file path of the markup document, and the resType is a parameter indicating an attribute of the markup document.

19. (Previously presented) The apparatus of claim 15, wherein the buffer manager informs the content decoder of the buffering state of the markup document using the API.

20. (Previously presented) The apparatus of claim 14, further comprising a content decoder to interpret the markup document;

wherein the buffer manager deletes the markup document from the buffer in response to a discard signal output from the content decoder.

21. (Previously presented) The apparatus of claim 20, wherein the content decoder generates the discard signal using a discard API.

22. (Previously presented) The apparatus of claim 14, wherein the interactive mode is a mode in which the AV data is displayed in a display window defined by the markup document;

the apparatus is selectively operable in the interactive mode in which the AV data is displayed in the display window defined by the markup document, and a non-interactive video mode in which the AV data is displayed in the same manner as AV data recorded on a standard DVD; and

the user of the apparatus selects between the interactive mode and the non-interactive video mode.

23. (Previously presented) An apparatus for recording and/or reproducing audio video (AV) data using a markup document in an interactive mode selected by a user of the apparatus before the apparatus reproduces any of the AV data, comprising:

- an AV buffer to buffer the AV data;

- an AV reproduction engine to decode the AV data;

- an enhanced navigation (ENAV) buffer to preload the markup document before the apparatus reproduces any of the AV data to enable the apparatus to reproduce the AV data in the interactive mode selected by the user;

- an ENAV engine to interpret the markup document, and identify buffering state information of the markup document in response to a report signal, the buffering state information being used by the apparatus in reproducing the AV data in the interactive mode selected by the user; and

- an I/O manager to obtain the markup document;

wherein:

- the ENAV engine generates the report signal using an application program interface (API); and

- the report signal is used by the ENAV engine to verify whether the markup document has been successfully preloaded into the ENAV buffer, whether the markup document cannot be read due to an error, and whether the markup document is being read.

24. (Previously presented) The apparatus of claim 23, wherein the I/O manager uses a blocked I/O method to obtain the markup document from a data storage medium, and uses an unblocked I/O method to obtain the markup document from a network.

25. (Previously presented) A method of reproducing AV data in an interactive mode using a markup document, the method comprising:

- buffering the markup document to preload the markup document;

- generating a report signal used to identify a buffering state of the markup document using an application program interface (API);

- using the report signal to verify whether the markup document has been successfully preloaded, whether the markup document cannot be read due to an error, and whether the markup document is being read; and

outputting buffering state information of the markup document in response to the report signal.

26.-27. (Canceled)

28. (Previously presented) The method of claim 25, wherein the API comprises a file path and an attribute of the markup document as parameters.

29. (Previously presented) The method of claim 25, wherein the API comprises an [obj].isCached(URL, resType) API that generates the report signal, where the URL is a parameter indicating a file path of the markup document, and the resType is a parameter indicating an attribute of the markup document.

30. (Previously presented) The method of claim 25, wherein the outputting of the buffering state information comprises returning a value of 0 in response to the markup document being successfully preloaded, returning a value of 1 in response to the markup document not being successfully preloaded, and returning a value of 2 in response to the markup document still being preloaded.

31. (Previously presented) The method of claim 25, further comprising reproducing the AV data in the interactive mode using the preloaded markup document.

32. (Previously presented) A method of managing a markup document for use in reproducing AV data in an interactive mode, the method comprising:

buffering the markup document to preload the markup document in response to a fetch signal;

outputting a buffering state of the markup document in response to a report signal;

staging the markup document for decoding in response to a retrieve signal; and

deleting the markup document in response to a discard signal.

33. (Previously presented) The method of claim 32, further comprising issuing a response indicating whether a command to preload the markup document included in the fetch signal has been successfully transmitted.

34. (Previously presented) The method of claim 32, wherein the outputting of the buffering state comprises returning a signal indicating whether preloading of the markup document has been completed.

35. (Previously presented) A method of managing a markup document for use in reproducing AV data in an interactive mode, the method comprising:
generating a fetch signal to preload the markup document;
generating a report signal to determine a buffering state of the markup document;
generating a retrieve signal to stage the markup document for decoding; and
generating a discard signal to delete the markup document.

36. (Previously presented) The method of claim 35, further comprising generating a release signal in response to the markup document no longer being presented.

37. (Previously presented) The method of claim 32, wherein the outputting of a buffering state of the markup document in response to a report signal comprises:
generating the report signal using an application program interface (API);
using the report signal to verify whether the markup document has been successfully preloaded, whether the markup document cannot be read due to an error, and whether the markup document is being read; and
outputting the buffering state of the markup document in response to the report signal.

38. (Previously presented) The method of claim 35, wherein the generating of a report signal to determine a buffering state of the markup document comprises:
generating the report signal using an application program interface (API);

using the report signal to verify whether the markup document has been successfully preloaded, whether the markup document cannot be read due to an error, and whether the markup document is being read; and
outputting the buffering state of the markup document in response to the report signal.